BG/L Systems Software Discussion

Summary of discussion

BG/L Consortium Kickoff

Systems Software

- ☐ Everything except applications
  - ■ Scheduler, queue manager, process manager, accounting system
  - ■ Conventional compilers: C, Fortran-9x, C++
  - ■ More languages: Java, UPC, Co-Array Fortran
  - ■ Helpful scripting languages: Perl, Python
  - ■ Basic parallel libraries: MPI, ARMCI, GA
  - ■ Higher-level libraries: ScaLAPACK, PETSc, HDF5, GA
  - ■ Math libraries
  - ■ Other things that applications need but don't include
- ☐ Described in Section 9 of *An Overview of BG/L*

Improvements to the document

- ☐ Explain VNM better – always one thread/processor independent of node
- ☐ We need a clear definition of what the CNK can and cannot do
- ☐ A lot of interest in more details of MPI implementation – how are things supported?
- ☐ Examples of how to code for double-FPU generation by compiler

New requirements

- Clear requirement for dynamic linking, because applications determine at run-time which libraries are necessary – Could be done by user-mode run-time library

- Suggestion that we expand our collaborations in performance tools – in particular Al Malony

- Guang Gao mentioned the need to access hardware performance monitors at a level below PAPI – may need to expose lower level interfaces

- People raised concerns about byte/flop ratio on a single link and also sharing of resources by both processors – need application studies to investigate behavior of BG/L in those circumstances

New requirements

- ☐ Clear interest from community on high-performance math libraries
- ☐ Issue of access to IBM source code has been raised
- ☐ Issue of MPI-I/O – which file system(s) will be supported

Questions

1. List of high priority systems software items for porting

   MPI-I/O, math libraries, performance tools, dynamic linking

2. Needs for accelerating systems software for BG/L

   Examples of coding and means to access system software

3. Coordination mechanisms

   Mailing list, "Source Forge" for BG/L

4. Stuff needed from IBM to facilitate systems software porting and development

   Documentation, access to experts and hardware

More Questions

5. Strategies for encouraging more participation in systems software development

   Further meetings, sharing of results (good and bad)

6. Prospect of an all open source systems software stack

   Consortium can develop its own stack (e.g., own LWK, Open64)

7. Requirements for additional file systems, scheduling options

   GPFS, Lustre, PVFS

8. Putting systems on the Grid? External I/O.

   Need more discussion/understanding

9. SS infrastructure for using the systems as a test bed

   Yes

The end